

Fourth Communications

Lab2_organization

Sec (1):

1- Asmaa Mohamed Kamal

2- Asmaa Abdelmonem Eyssa

3- Asmaa Shaban Mahmoud

4- Basma Gamal

ANSWER QUESTION (1):

Add ACC to the MDR

Type of Microinstruction: **Arithmetic**

name	type	source1	source2	destination	overflowBit	carryBit
acc+A->acc	ADD	acc	A	acc	halt-bit	(none)
acc+mdr->...	ADD	acc	mdr	acc	halt-bit	(none)
acc-mdr->...	SUBTRACT	acc	mdr	acc	halt-bit	(none)
acc/mdr->...	DIVIDE	acc	mdr	acc	halt-bit	(none)

New Delete Duplicate

Help OK Cancel

test ACC if not zero , Skip the next microinstruction

Type of Microinstruction: **Test**

name	register	start	numBits	comparison	value	omission
if(acc!=0)s...	acc	0	16	NE	0	1
if(acc>=0)s...	acc	0	16	GE	0	1

New Delete Duplicate

Help OK Cancel

ANSWER QUESTION (2):

Microinstruction Write

The screenshot shows the 'Edit machine instructions' window. On the left, a table lists microinstructions with their names, opcodes, and field lengths. The 'write' instruction is selected. In the center, the 'write's implementation' box contains the code: 'output-acc->int' followed by 'End'. On the right, a list of 'Existing micros' includes TransferRtoR, TransferAtoR, TransferRtoA, Set, Test, Increment, Arithmetic, Shift, Branch, Logical, Decode, MemoryAccess, IO, and SetCondBit. At the bottom are buttons for 'New', 'Delete', 'Duplicate', 'Help', 'OK', and 'Cancel'.

name	opcode	fieldLengths
Add	C	4 12
stop	0	16
load	1	4 12
store	2	4 12
read	3	4 (12)
write	4	4 (12)
add	5	4 12
subtract	6	4 12
divide	8	4 12
jump	9	4 12
jmpz	A	4 12
jmpn	B	4 12

write's implementation

```
output-acc->int
End
```

Existing micros

- TransferRtoR
- TransferAtoR
- TransferRtoA
- Set
- Test
- Increment
- Arithmetic
- Shift
- Branch
- Logical
- Decode
- MemoryAccess
- IO
- SetCondBit

Buttons: New, Delete, Duplicate, Help, OK, Cancel

Microinstruction Divide

The screenshot shows the 'Edit machine instructions' window. On the left, a table lists microinstructions with their names, opcodes, and field lengths. The 'divide' instruction is selected. In the center, the 'divide's implementation' box contains the code: 'ir(4-15)->mar', 'Main[mar]->mdr', 'acc/mdr->acc', followed by 'End'. On the right, a list of 'Existing micros' includes TransferRtoR, TransferAtoR, TransferRtoA, Set, Test, Increment, Arithmetic, Shift, Branch, Logical, Decode, MemoryAccess, IO, and SetCondBit. At the bottom are buttons for 'New', 'Delete', 'Duplicate', 'Help', 'OK', and 'Cancel'.

name	opcode	fieldLengths
Add	C	4 12
stop	0	16
load	1	4 12
store	2	4 12
read	3	4 (12)
write	4	4 (12)
add	5	4 12
subtract	6	4 12
divide	8	4 12
jump	9	4 12
jmpz	A	4 12
jmpn	B	4 12

divide's implementation

```
ir(4-15)->mar
Main[mar]->mdr
acc/mdr->acc
End
```

Existing micros

- TransferRtoR
- TransferAtoR
- TransferRtoA
- Set
- Test
- Increment
- Arithmetic
- Shift
- Branch
- Logical
- Decode
- MemoryAccess
- IO
- SetCondBit

Buttons: New, Delete, Duplicate, Help, OK, Cancel

Microinstruction Halt(stop)

The screenshot shows the 'Edit machine instructions' dialog box. On the left, a table lists microinstructions. The 'stop' instruction is selected, showing its opcode as '0' and field lengths as '16'. Below the table are 'New', 'Delete', and 'Duplicate' buttons. The main area is titled 'stop's implementation' and contains the code: 'set-halt-bit' followed by 'End' on the next line. To the right, a list of 'Existing micros' includes TransferRtoR, TransferAtoR, TransferRtoA, Set, Test, Increment, Arithmetic, Shift, Branch, Logical, Decode, MemoryAccess, IO, and SetCondBit. Between the implementation and existing micros are '<<insert<<' and '>>delete' buttons. At the bottom are 'Help', 'OK', and 'Cancel' buttons.

name	opcode	fieldLengths
Add		C 4 12
stop	0	16
load		1 4 12
store		2 4 12
read		3 4 (12)
write		4 4 (12)
add		5 4 12
subtract		6 4 12
divide		8 4 12
jump		9 4 12
jmpz		A 4 12
jmpn		B 4 12

stop's implementation

```
set-halt-bit
End
```

Existing micros

- TransferRtoR
- TransferAtoR
- TransferRtoA
- Set
- Test
- Increment
- Arithmetic
- Shift
- Branch
- Logical
- Decode
- MemoryAccess
- IO
- SetCondBit

Buttons: New, Delete, Duplicate, Help, OK, Cancel

Microinstruction Jmpn

The screenshot shows the 'Edit machine instructions' dialog box. The 'jmpn' instruction is selected in the table on the left, showing its opcode as 'B' and field lengths as '4 12'. The main area, titled 'jmpn's implementation', contains the code: 'if(acc>=0)skip-1', 'ir(4-15)->pc', and 'End' on the next line. The 'Existing micros' list on the right is identical to the previous screenshot. The '<<insert<<' and '>>delete' buttons are also present. At the bottom are 'Help', 'OK', and 'Cancel' buttons.

name	opcode	fieldLengths
Add		C 4 12
stop	0	16
load		1 4 12
store		2 4 12
read		3 4 (12)
write		4 4 (12)
add		5 4 12
subtract		6 4 12
divide		8 4 12
jump		9 4 12
jmpz		A 4 12
jmpn	B	4 12

jmpn's implementation

```
if(acc>=0)skip-1
ir(4-15)->pc
End
```

Existing micros

- TransferRtoR
- TransferAtoR
- TransferRtoA
- Set
- Test
- Increment
- Arithmetic
- Shift
- Branch
- Logical
- Decode
- MemoryAccess
- IO
- SetCondBit

Buttons: New, Delete, Duplicate, Help, OK, Cancel

ANSWER QUESTION (3):

